

## REMARKS

### CLAIM OBJECTIONS

Claims 8, 13 and 18 were objected to for informalities and have been appropriately amended.

### REJECTIONS UNDER 35 U.S.C. § 112

Claims 6-20 were rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Applicant points at least to paragraph [0035] to support assigning a register class based at least in part on a conjunctive forward dataflow analysis. Support need only be ascertainable by one of ordinary skill in the art – verbatim language supporting the claim is not required.

Furthermore, Applicant points at least to paragraph [0030] to support assigning a register assignment map at both a block entry and a block exit.

Claims 6-20 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Applicant has amended the claims in question to make them clearer.

### REJECTIONS UNDER 35 U.S.C. § 103

Claims 6-20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 4,961,141 issued to Hopkins et al. (*Hopkins*) in view of Esparza et al: “Efficient algorithms for pre\* and post\* on interprocedural parallel flow graphs,” ACM,

2000 (*Esparza*). Applicant submits claims 6-20 are not obvious in view of *Hopkins* and *Esparza* for at least the reasons set forth below.

Claim 6 recites, in part, the following:

assigning a second register class to the at least one symbolic register based at least in part on a linear conjunctive forward dataflow analysis that iterates a basic block of instructions in a compilation unit only once;

Independent claims 11 and 16 recite similar limitations. The Office action correctly concedes the *Hopkins* fails to disclose conjunctive forward dataflow analysis that iterates a basic block of instructions only once. *Esparza* was cited to cure the deficiencies of *Hopkins*. However, *Esparza* does not expressly disclose iterating a basic block of instructions *only once*, as recited in the independent claims. Furthermore, it is not inherent in *Esparza* to iterate a basic block of instructions only once. For example, the cited portion of *Esparza* simply discusses a special form of a reduction clause containing two atoms with the same variable. See page 8, section 5 of *Esparza*. Whether or not *Esparza* discloses conjunctive dataflow analysis (which is not conceded given that *Esparza* discusses “interprocedural” dataflow analysis rather than conjunctive dataflow analysis – see page 1, section 1, first paragraph), it does not necessarily flow (as required by MPEP § 2112) from *Esparza* to assign a second register class to the at least one symbolic register based at least in part on a linear conjunctive forward dataflow analysis that iterates a basic block of instructions in a compilation unit only once. Thus, *Esparza* fails to cure the deficiencies of *Hopkins*.

Even assuming, for the sake of argument, that *Esparza* did disclose iterating each basic block of instructions only once, combining *Esparza* with *Hopkins* in this sense

would change the principle of operation of *Hopkins*. *Hopkins* specifically discloses and requires, per Fig. 2 and col. 3, lines 33-44, at least two passes (or iterations) are made to a program in which “fix up” code is inserted. Thus, even if *Esparza* disclosed iterating a basic block of instructions only once (which Applicant contends it does not), combining *Esparza* with *Hopkins* would result in an inoperable combination. Thus, in either scenario, the combination of *Hopkins* and *Esparza* fails to render claims 6, 11 and 16 obvious for at least the reason that *Hopkins* and *Esparza* fail to disclose conjunctive forward dataflow analysis that iterates a basic block of instructions in a compilation unit only once.

Additionally, claim 6 recites, in part, the following:

moving existing register class fixups for the assignment of the second register class to a different location and removing unnecessary register class fixups, the moving and removing to reduce the register class fixups, wherein register class fixups are instructions inserted into a program in response to one or more register class assignment modifications; and

Column 3, lines 33-44 of *Hopkins* is cited as disclosing moving register class fixups. Per *Hopkins*, “fix up” code is merely inserted into a program to move a value from one space to another. Again, for clarity, *Hopkins* discusses inserting “fix up” code and not moving “fix up” code that has already been inserted. In contrast, Applicant’s claims recite moving fixups to a different location (i.e., moving fix up code that has already been inserted). As discussed in Applicant’s specification in paragraph [0038], moving fixups to a different location (as opposed to simply inserting them) is a form of optimization that can reduce the total number of fixups needed. *Hopkins* clearly does not disclose such an optimization. *Esparza* was not cited as disclosing, nor does it disclose,

such an optimization. Thus, *Esparza* again fails to cure the deficiencies of *Hopkins*. Therefore, once again, Applicant submits the independent claims are not obvious in view of *Hopkins* and *Esparza*.

Claims 7-10 depend from claim 6. Claims 12-15 depend from claim 11. Claims 17-20 depend from claim 16. Given that dependent claims necessarily include the limitations of the claims from which they depend, Applicant submits claims 7-10, 12-15 and 17-20 are not obvious for at least the same reasons claims 6, 11 and 16 are not obvious.

CONCLUSION

For at least the foregoing reasons, Applicant submits that the rejections have been overcome. Therefore, claims 6-20 are in condition for allowance and such action is earnestly solicited. The Examiner is respectfully requested to contact the undersigned by telephone if such contact would further the examination of the present application.

Please charge any shortages and credit any overcharges to our Deposit Account number 02-2666.

Respectfully submitted,  
**BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP**

April 7, 2009

Date

/Jared S. Engstrom/

Jared S. Engstrom

Reg. No. 58,330

Attorney for Applicant

1279 Oakmead Parkway  
Sunnyvale, CA 94085-4040  
Telephone: 503-439-8778

I hereby certify that this correspondence is being submitted electronically via EFS Web on the date shown below:

Date: 4/07/2009

/Heidi Yerton/

Heidi Yerton